MicroCART: IRP Presentation

MicroCART: Microprocessor Controlled Aerial Robotics Team

Daniel Zaucha, Jonah Upah, Ryan Lowe, Yi Hang Ang

Advisor/Client: Dr. Phillip Jones
Team Website: https://sdmay25-32.sd.ece.iastate.edu

Project Overview

- Ongoing project revolving around the development of a quadcopter platform utilized by Iowa State students and controls systems researchers
- CPRE 4880 Lab 4
- Fixing known issues in the MicroCART application.
- Refine documentation
- Enhance data collection capabilities and data representation on the GUI.

Intended Users

- CPRE 4880 Students
  - Complete Lab 4.
- Project Advisor / Class TA
  - Need to be able to assist with Lab 4.
- Prospective ISU Students
  - Perform demos for high school students.
- Successor Project Teams
  - Future teams can brought up to speed with ease.

Technologies Utilized

- Bitcraze CrazyFlie Drone
  - Micro quadcopters manufactured by Bitcraze.
  - Drones used by CPRE 4880 students to complete Lab 4.

- Virtual Machine
  - Project environment inside a virtual machine.
  - Used as a development environment for Lab 4.

- Arduino
  - Used an Arduino for the test stand tracker.

CrazyFlie mini-quadcopter

Project Goals

- Overall goal for our project this year has been focused solely on the improvement of Lab 4.
  - In previous years, this lab has frequently experienced issues such as VM/GUI lagging and crashing.
    - Mismanagement of system resources.
      - Unreliable connection to drone.
  - Improve the performance of the VM/GUI.
    - Limit the amount of crashes and non user-related problems.

Project Goals

- Utilize the test stand sensor
  - Enhance data collection capabilities and data representation on the GUI.
    - Functions as an external sensor relative to the CrazyFlie's onboard sensors
    - Allows external validation of the onboard sensor data

Similar looking Sensor [3]
Part Number of test stand sensor: MA3-P12-125-2

Project Goals

- Refining Documentation
  - Revise lab instructions and terminology.
    - Include information about current implementation.
    - Correcting inaccurate documentation of key terms.
  - Update the knowledge base for future teams.
    - Highlight key changes implemented by the team.
      - Test stand documentation
    - Provide instructions for editing CrazyFlie firmware.
    - Starting points in the Git Repo.

Project Requirements

- CPRE 4880 Lab 4
  - Students should be able to complete the lab without frequent crashes or performance issues.
  - Students should have updated information available to them through either the lab documentations or CPRE 4880 Wiki page.
  - Students should have access to functional lab equipment
    - A Virtual Machine with a fully operational lab environment should be deployed on lab computers.
    - Sufficient functional drones should be available for use

Project Requirements

- Documentation
  - Any changes made to the MicroCART infrastructure should be documented for future team reference.
  - Keep all information up to date by revising anything outdated or inaccurate.
  - Create/Update documentation to enable future teams to get up to speed within 4-6 weeks

Risks

- Deployment of the VM for Lab 4
  - No changes to the VM can be made after deployment.
- Test stand tracker
  - If not dismounted from VM, caused an entire system crash.
- Maintaining the health of the lab equipment
  - Ideally want at least 12 working quadcopters and radio dongles for lab use.

Mitigation

- Deployment of the VM for Lab 4
  - Acted as TAs during lab periods and helped fix any issues that came up during lab.
- Test stand tracker
  - Emphasized in the lab manual and Discord server for students to dismount before shutting down GUI
- Maintaining the health of the lab equipment
  - Constantly updated a health status sheet for the quadcopters
  - Purchased broken parts to fix drones / had assistance from ETG to fix broken parts

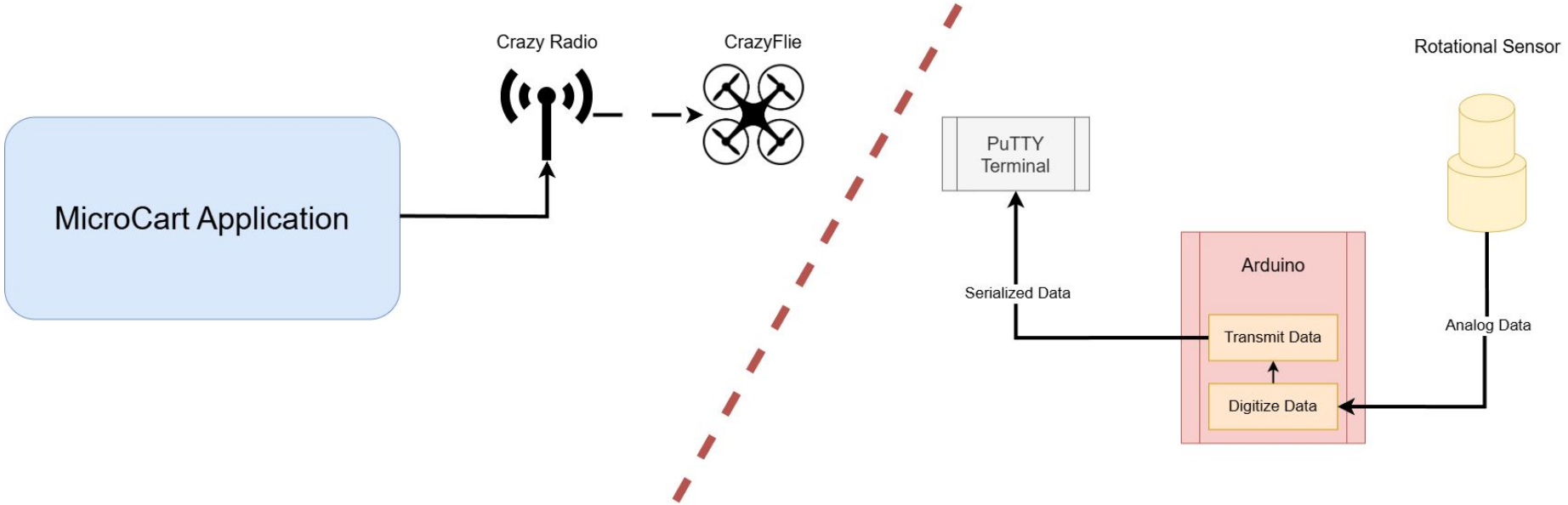
Continual Maintenance Resource/Cost Estimation

- Expected: 12 sets of operational lab equipment
- Single Set:
  - Crazyflie
  - 2x Batteries
  - Crazyradio

Parts	Amount	Cost (Ignoring Taxes)[1][2]
Crazyradio 2.0	x7	$43/unit $301.00
250mAh LiPo Battery	x13	$6.50/unit $84.50
USB Extension Cord	x8	$11.39/2 units $45.56
Total Cost: $431.06	Remaining Budget: $68.94	Real and Estimate Difference: Cost $203.86 more than Estimate

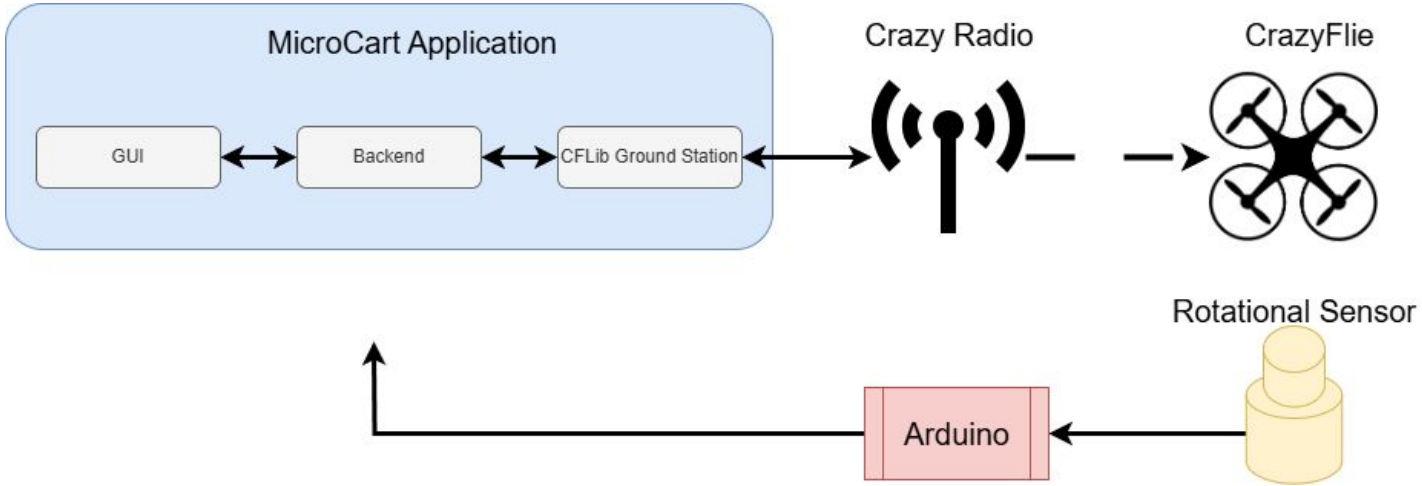
Initial Design

- Test stand Data Pipeline



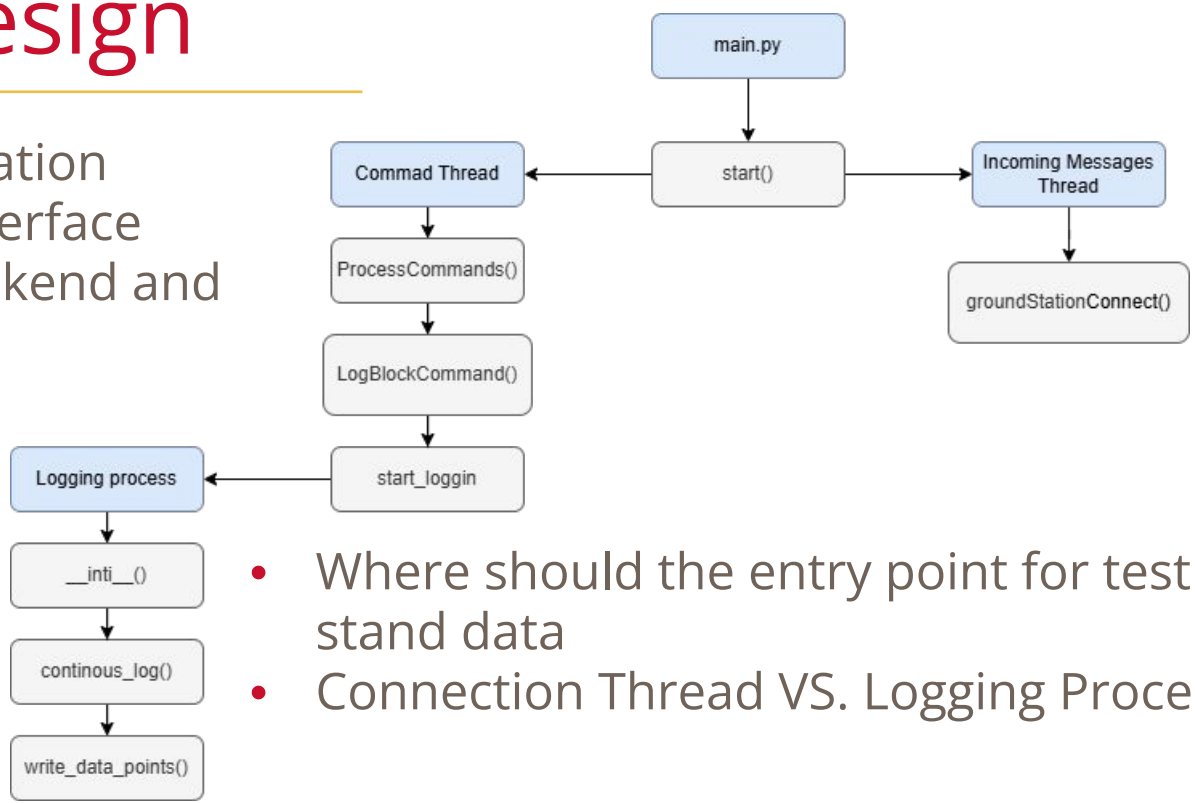
Initial Design

- Entry Point for Test stand Data



Detailed Design

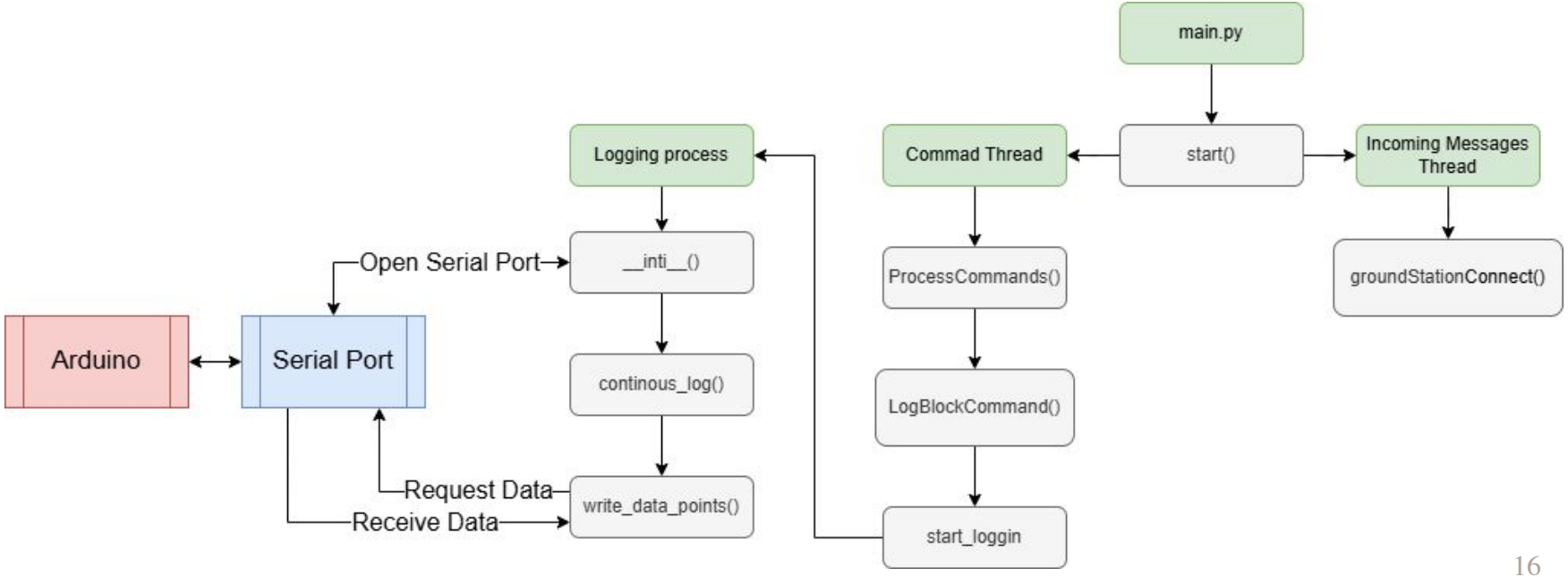
- CFLib Ground station
- This acts as a interface between the backend and the CFLib.



- Where should the entry point for test stand data
- Connection Thread VS. Logging Process

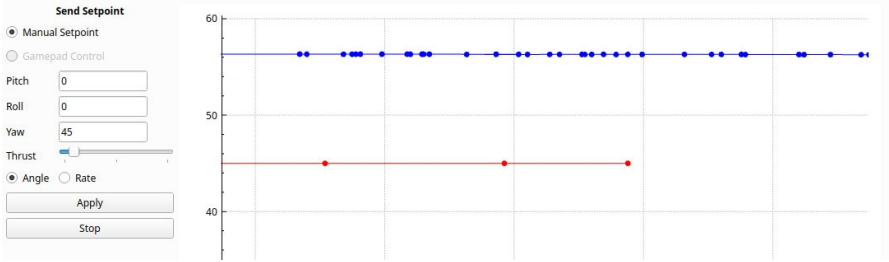
Detailed Design

- Test Stand Integration



Detailed Design

- Optimization of Application
  - Usage of busy waits
  - Duplication of logging process




```
  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
 2707 bitcraze  20   0    6284   1900   1736 R  77.3   0.0  91:39.59 ./BackEnd
 2826 bitcraze  20   0  797032  23948   4700 R  54.3   0.1   0:05.73 python3 main.py 50
 2675 bitcraze  20   0  797032  34244  15020 S  45.0   0.2  43:04.98 python3 main.py 50
 2824 bitcraze  20   0  797032  23940   4700 S  41.3   0.1   0:11.82 python3 main.py 50
 2741 bitcraze  20   0  797032  23676   4436 S  34.0   0.1  62:49.64 python3 main.py 50
 2738 bitcraze  20   0  797032  23676   4436 S  33.7   0.1   4:36.41 python3 main.py 50
 2708 bitcraze  20   0 1713932 175776  92804 S  32.3   0.9   2:30.54 ./GroundStation
 2735 bitcraze  20   0  731496  23624   4436 S  29.3   0.1  64:46.72 python3 main.py 50
 2742 bitcraze  20   0  797032  23664   4428 S  18.3   0.1  61:56.81 python3 main.py 50
 1050 root      20   0 1196548 127260  43088 S   4.3   0.7
 1633 bitcraze  20   0  388112  39612  30028 S   1.0   0.2   0:30.60 /usr/bin/xfce4-terminal
 1454 bitcraze  20   0 1263276  93176  66512 S   0.3   0.5   0:12.97 xfwm4 --replace
 1470 bitcraze  20   0  229092  20508  14736 S   0.3   0.1   0:00.31 xfsettingsd
 1703 bitcraze  20   0  494012  70264  38204 S   0.3   0.4   0:05.90 /usr/bin/gedit --gappli
    1 root      20   0  167308  11464   8564 S   0.0   0.1   0:02.05 /sbin/init splash
    2 root      20   0       0      0      0 S   0.0   0.0   0:00.01 [kthreadd]
```

Detailed Design

- Optimization of Develop Environment
    - VM resources
    - Limit LLVMpipes
    - Deployment process

Challenges

- The whole group had problems getting brought up to speed with the project
  - Did not have enough meeting times that worked for everyone 1st semester.
- Git repository has ~8 years of development, making it difficult to grasp.
- Gaps in documentation from earlier MicroCART work.
- Difficulties with Hardware

Solutions

- Creating handover documents
  - Making it easier for next year's team to get brought up to speed
- Cleaning up the Git repository to make it more clear what parts are relevant
- Posting videos on navigating the Git repository
- Issues, causes, and solutions documented

Testing & Feedback

- Our testing of the entire system was done by completing Lab 4 from a students perspective.
- Testing of the test stand was done by comparing the test stand data with the on-board drone sensor

- As part of Lab 4, we also ask students for feedback to help with future implementations:
  - Some TODO's in part 2 need to be more clear for what it wants.
  - Have a way to test part 2 without having to flash the drone.
  - May be too much documentation, for part 1 especially.
  - Change battery connector so that the wires do not break.

Results

- Managed to reduce CPU utilization from 100% to around 60%-70%
  - Allowed for much smoother logging and stability of the VM as a whole
- Successfully implemented a test stand tracker before the deployment of Lab 4

Results

- Had a successful deployment of Lab 4
  - “Smoothest ever deployment of the lab”, according to Dr. Jones, Client & CPR E 4880 Teacher
  - All lab groups were able to complete the lab successfully
- Collected feedback from students to pass on to next years team for their reference
- Made changes to the Lab 4 manual to improve readability

Conclusion

- Individual Contributions
  - Jonah: Arduino integration, Test stand logging, VM/GUI performance improvements
  - Yi: Test stand logging, Lab 4 firmware changes and implementations, VM/GUI performance improvements
  - Ryan: Lighthouse research, Lab 4 testing, troubleshooting, documentation
  - Daniel: Bug testing/fixing/recording, Equipment maintenance, Lighthouse system